

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problems Mailbox.**

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2000-11199

(P2000-11199A)

(43)公開日 平成12年1月14日 (2000.1.14)

(51)Int.Cl.<sup>7</sup>

G 0 6 T 13/00

識別記号

F I

テーマコード(参考)

G 0 6 F 15/62

3 4 0 D 5 B 0 5 0

審査請求 未請求 請求項の数 6 O.L. (全 10 頁)

(21)出願番号 特願平10-170863

(22)出願日 平成10年6月18日 (1998.6.18)

(71)出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(71)出願人 592215103

株式会社ソニー木原研究所

東京都品川区東五反田1丁目14番10号

(72)発明者 大戸 康紀

東京都品川区東五反田1丁目14番10号 株式会社ソニー木原研究所内

(74)代理人 100091546

弁理士 佐藤 正美

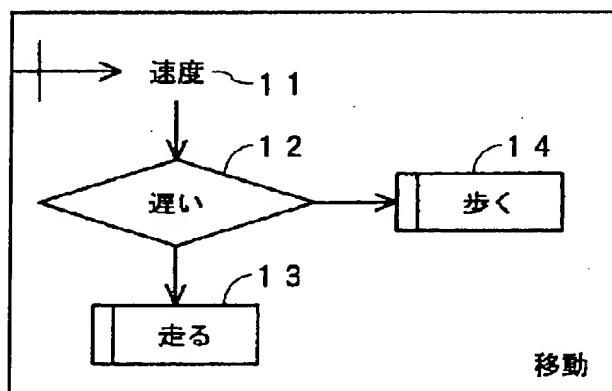
最終頁に続く

(54)【発明の名称】 アニメーションの自動生成方法

(57)【要約】

【課題】 アニメーションデータの作成において、個々のアニメーション間の移行をスムーズに行う。

【解決手段】 個々のアニメーションを行動オブジェクトとして持つ。作成するアニメーションをそれらの状態遷移として扱う。状態遷移を行うタイミングはイベントとして持つ。



→ 入力

◇ 判断

□ 行動オブジェクト

**【特許請求の範囲】**

【請求項1】アニメーションデータの作成において、個々のアニメーションを行動オブジェクトとして持ち、作成するアニメーションをそれらの状態遷移として扱うようにしたアニメーションの自動生成方法。

【請求項2】請求項1に記載のアニメーションの自動生成方法において、

上記状態遷移を行うタイミングをイベントとして持つようにしたアニメーションの自動生成方法。

【請求項3】請求項2に記載のアニメーションの自動生成方法において、

上記状態遷移の管理と、行動オブジェクトの管理とを独立に行うようにしたアニメーションの自動生成方法。

【請求項4】請求項3に記載のアニメーションの自動生成方法において、あるオブジェクト状態がイベントを受け取るタイミングを制限しないとともに、その同期タイミングを調節できるようにしたアニメーションの自動生成方法。

【請求項5】請求項4に記載のアニメーションの自動生成方法において、

上記イベントとして、時間、割り込み、他のオブジェクトからの出力、環境情報などを設定するようにしたアニメーションの自動生成方法。

【請求項6】請求項5に記載のアニメーションの自動生成方法において、

アニメーションの対象として2次元データあるいは3次元データを使用するようにしたアニメーションの自動生成方法。

**【発明の詳細な説明】****【0001】**

【発明の属する技術分野】この発明は、アニメーションの自動生成方法、特に2次元あるいは3次元形状のアニメーションの自動生成において、シーンの移り変わりをスムーズにする技術に関する。

**【0002】**

【従来の技術】ゲーム機やプレゼンテーションなどにおいて、アニメーションの作成や再生が幅広く利用されている。

【0003】図21は、キーフレームアニメーションの生成の様子を示すもので、アニメーション製作者がキーフレームとしてフレーム1、3を指定することにより、これらキーフレーム1、3から、これらの間の補間フレーム2を生成する。そして、このような作業を、指定した時間区分について行うことによって、歩く、跳ぶなどの個々のアニメーションを生成し、さらに、個々のアニメーションを時間軸に沿って並べていくことにより全体のアニメーションとしている。

【0004】なお、これらの方法を用いたアニメーション作成ツールとして、Macromedia社のDirectorなどが知

られている。

**【0005】**

【発明が解決しようとする課題】ところが、上述のように、従来のアニメーション作成においては、時間的に短いアニメーションをつないで全体のアニメーションを作成しているので、ユーザ（ゲームのプレーヤなど）とのインタラクションを行う場合においても、短いアニメーションの動きが終了するまで次のアニメーションへ移っていくことができなかった。また、これを回避するために、ユーザからの入力を想定して個々の遷移に応じたアニメーションを作成していく必要があった。

【0006】例えば、図22は、従来方法によるアニメーションの切り換えの様子を示すもので、「歩く」の場合には、アニメーション生成は、フレーム1→フレーム2→フレーム3の流れであるが、ユーザがボタンを押すなどのイベントを発生させることにより「歩く」から「跳ぶ」に切り換わるときには、フレーム1→フレーム4→フレーム5の流れへと切り換わる。つまり、アニメーションの切り換え点は、共通のフレーム1の位置に制限される。

【0007】このため、各アニメーションをつなぐと、そのつなぎ目がスムーズでなかったり、あるアニメーションを行っている間は、ユーザからの入力をアニメーションに反映させることができなかったりする。

【0008】したがって、イベントに対するアニメーションの変化と、スムーズにつながれたアニメーションの作成とを同時に行なうことはできず、例えば図23に示す格闘ゲームでは、キャラクタが「技」のアニメーションに入ると、そのアニメーションの中に設けられた「弾き」、「よけ」、「防御」などのコマンド受け付けのタイミングになるまで、アニメーションが続行される。

【0009】また、「受け」に対するアニメーションに入るとなどに、アニメーションのつなぎ目に乱れが見受けられ、さらに複雑に入り込んだ攻防については、タイミング設計が困難であった。

【0010】この発明は、このような問題点を解決しようとするものである。

**【0011】**

【課題を解決するための手段】このため、この発明においては、アニメーションデータの作成において、個々のアニメーションを行動オブジェクトとして持ち、作成するアニメーションをそれらの状態遷移として扱うようにしたアニメーションの自動生成方法とするものである。したがって、イベントにより個々のアニメーション間の移行が行われ、その移行はスムーズに表現される。

**【0012】**

【発明の実施の形態】図17は、この発明によるアニメーションの生成例を示す。すなわち、例えば、「歩く」から「跳ぶ」へ移行する場合、歩くというアニメーションの途中から跳ぶというアニメーションへのつなぎのア

アニメーション、例えば歩いているところから片足を軸にして跳ぶ動作に入るまでのアニメーションを自動的に生成し、アニメーションが急に変わることなくスムーズに移行していくような表現を可能にする。

【0013】つまり、「歩く」、「跳ぶ」などといった個々のアニメーション状態（時間軸に沿った一連の動き）を「行動オブジェクト」として保持し、全体のアニメーションはそれらの状態遷移として考える。そして、ある行動オブジェクトから別の行動オブジェクトには、イベントによって移行する。

【0014】図1は、「移動」オブジェクトにおける状態遷移の例を示す。そして、符号11はオブジェクトの外からのイベントであり、オブジェクト内部に設けた判断機構12によってイベントを振り分けることにより、オブジェクト内部に設定した、行動オブジェクト13あるいは14を実行する。

【0015】このように、行動オブジェクトを階層化することにより動作の抽象化を行うこともできる。行動オブジェクトは、状態遷移とは独立して動いており、オブジェクト間の相互作用によって、モデル全体に対するアニメーションを生成するので、各行動オブジェクトはその状態が完結していないくとも、イベントを受け付けることができる。

【0016】次に、この発明のアニメーション自動生成方法について、アニメーション作成エディタを利用して説明する。なお、エディタでは、「歩く」、「跳ぶ」というアニメーションを使用し、それらをどのように合成して新しい「歩きから跳ぶ」というアニメーションを生成するかについて説明する。また、2つのアニメーション間のみならず、複数のアニメーションの合成により、複雑なアニメーションを生成する方法についても説明を行う。

【0017】図2は、この発明を適用したアニメーションの自動作成装置を示す。この装置において、一連のアニメーションを設計、製作する場合には、マウス、タブレット、キーボード、ジョイスティックなどの入力装置21が使用され、CPU22において、個々の行動オブジェクト内のアニメーション、同期用のイベント発生ポイントの作成などが行われ、これらはメモリ23に記憶される。

【0018】そして、例えば全体を通したアニメーションの確認時、CPU22によりメモリ23に記憶されている個々のアニメーションが合成され、ディスプレイ、ビデオ機器、プリンタなどの出力装置24に出力される。また、一連の行動オブジェクトやその関係図を作成したときには、それらがHDD、CD-R、DVDなどの大容量記憶装置25に出力される。

【0019】個々の行動オブジェクトからの全体アニメーションの生成は、アニメーションを実際にを行うシステム、例えばゲーム機において、ジョイスティックなどの

入力イベントに対応して動的に行われる。

【0020】図3に、アニメーション生成に関するモジュール構成を示す。まず、ステップ31において、アニメーションを生成する上での素材となる個々のアニメーションや、オブジェクト間の関係などを作成する。次に、ステップ32において、ステップ31により作成した行動オブジェクトを用いて、ジョイスティックなどの入力装置からのイベントに対応したアニメーションを生成し、これをステップ33において、ディスプレイなどに出力する。

【0021】続いて行動オブジェクトの作成方法を示し、次に行動オブジェクト群から全体アニメーションの生成する方法について説明する。

【0022】図4は、行動オブジェクトの作成方法の流れを示す。すなわち、最初に、個々の行動オブジェクトに対してアニメーションを割り当てる（ステップ41）。この割り当ては、キーフレームアニメーションの場合には、例えば図5に示すような表示のもとで行われるもので、図5Aはプレビュー画面、図5Bは編集画面である。そして、編集画面（図5B）において、符号51はタイムラインを表す。そして、このタイムライン51は、アニメーションに必要な時間52で規格化したものを利用している。また、目盛りの表示は、位相値（=経過時間／アニメーションの長さ）と実際の経過時間との両方が用意されている。

【0023】そして、タイムライン51の上にキー53を設定していく。このキー53は、アニメーションのための特徴点の状態を示すものであり、骨格構造54の他にポリゴンの頂点などを設定することもできる。

【0024】こうして、設定したアニメーションは、プレビュー画面（図5A）において、フレーム61→62→63のように表示され、結果を確認することができる。また、アニメーションの開始および終了のタイミング55、56（図5B）を調整することによって、不要なアニメーションの重なりなどを制御することができる（ステップ42）。なお、上記のキーフレームアニメーションの他には、パスブラインアニメーションやモーションキャプチャなどのアニメーションについても扱うことができる。

【0025】次に、行動オブジェクト間の優先付けを行う（ステップ43）。これは、すでに存在している行動オブジェクトが、新たに生成されたオブジェクトと競合する場合の優先順位となるものである。図6は、行動オブジェクト間の優先順位付け画面の例を示すもので、図6Aは、行動オブジェクト間の全体的な優先順位付け画面を示す。なお、優先順位は相対的に指定する。

【0026】また、細かな設定を可能にするために、図6Bに示すように、特定のオブジェクト間の優先順位を指定することもできる。この図6Bでは、「歩く」、「走る」は「揺する」に対して、優先順位が高くなつて

いる。これは、「歩く」が実行されているときに、「揺する」を実行することができないことを示している。

【0027】図7に、個々の行動オブジェクト間の状態遷移図の編集画面を示す。行動オブジェクトは階層化することが可能であり、ボタン71によって、最大表示、ウィンドウ表示74、最小表示75の状態を取ることができる。また、符号72は、オブジェクト内で保持される状態変数であり、図7では、オブジェクト外部からのイベントを保持している。

【0028】このイベントは、さらに関数73によって処理されている。オブジェクト74とオブジェクト75とは平行関係を示しており、歩行時に右手と左手とを逆位相で同期させる場合などに指定する。また、状態変数76をオブジェクトの外に出力することにより、他のオブジェクトとの関係付けを行うことができる。そして、編集した結果は、図5Aのプレビュー画面によって確認をすることができる。

【0029】図8はアニメーションの合成方法の流れを示す。行動オブジェクトの生成は、例えば図9に示すように、状態遷移によって管理する。行動オブジェクトはそれぞれ開始点81を有し、現在の状態82が遷移していく。図9においては、81→82→83と変化していくことを示している。また、ある時点を取ると、図9に示すように、存在中（アニメーション生成中）の行動オブジェクトが、一つのみの場合、複数ある場合、ない場合（休止状態）などがある。そして、図10にも示すように、生成された行動オブジェクトは、現在進行中のオブジェクトと相互作用を通じて合成され、全体アニメーションを生成していく。

【0030】図11は、新規オブジェクトと既存のオブジェクトとの関係をフローチャートとして示す。すなわち、新規オブジェクトが生成すると、初めに既存のオブジェクトとの競合関係をチェックする。そして、競合関係がある場合には、互いの優先順位をチェックし、後発のオブジェクトの優先順位が高ければ、遷移アニメーションを自動生成する。

【0031】また、競合関係がなく、さらに協調関係を取るオブジェクトがある場合には、同期アニメーションを自動生成する。この自動生成するアニメーションには、そのオブジェクトの関係から、「遷移」、「独立」、「同期」がある。「独立」してアニメーションを生成する方法は、互いの関係を考慮しなくてよいため、従来手法と同様にアニメーションの生成を行う。

【0032】さらに、競合関係があり、かつ、優先順位によってオブジェクトの置き換えが起きる場合には、遷移アニメーションが自動生成される。この生成方法は、行動の種類によって、図12に示すように5通り存在する。

【0033】図13は、ポーズ間における遷移アニメーションの生成方法を示す（図12の符号aに対応）。ま

ず、経過時間に対する遷移時間の比として、それぞれのモデルに対する重みを求め、モデルの特徴点に対して荷重和として、モデル（合成結果）(2)を計算する。そして、これを開始時点から終了時点まで繰り返すことにより、遷移アニメーションを生成する。

【0034】図14は、ポーズからアニメーションへの遷移方法を示す（図12の符号bに対応）。まず、経過時間における遷移対象アニメーション中のフレーム(2)を求めた後、ポーズ間のアニメーションと同様にして、合成結果を計算する。そして、これを開始時点から終了時点まで繰り返すことにより、遷移アニメーションを生成する。また、同様にして、アニメーションからポーズへの遷移アニメーション（図12の符号bに対応）を求めることができる。

【0035】図15は、アニメーションからアニメーションへの遷移方法を示す（図12の符号dに対応）。まず、遷移開始アニメーション(1)～(3)に対して、経過時間におけるフレーム(2)を計算し、同様にして遷移終了アニメーション(6)～(7)からフレーム(4)を計算する。そして、この二つのフレーム間(2)→(4)について、ポーズ間の遷移と同様にしてアニメーションを求める。

【0036】図16は、周期的アニメーション間の遷移方法を示す（図12の符号eに対応）。このように周期的な行動の場合、位相のずれをなくすことが重要になる。このため、位相同期ポイントとして、0、1/4、1/2、3/4の4点、あるいは0、1/2の2点を用意し、位相を合わせた形での遷移アニメーションを生成する。図16においては、点(3)、(8)が1/4、点(4)、(9)が1/2、点(5)、(10)が3/4となっている場合である。

【0037】そして、遷移アニメーションの開始点を、その前後の位相同期点からの内分点(2)として表現し、これに対応する点を位相同期点からの内分点(7)として求める。次に、同期に必要な時間（位相幅）、図16では仮に1/2後としたときの位相同期ポイント(5)、(10)を求める、これらから遷移開始アニメーションとして(2)→(5)を求める、終了アニメーションとして(7)→(10)を求める。そして、最後に、アニメーションからアニメーションへの遷移と同様にして、遷移アニメーションを生成する。

【0038】図17は、歩くアニメーションを持つ行動オブジェクトと、跳ぶアニメーションを持つ行動オブジェクトとの間におけるアニメーションの合成例と、その遷移図を示す。また、図18は、タイミングを考慮した合成例と、その遷移図を示す。

【0039】図19は、平行オブジェクトに属するアニメーション(1)、(2)から、同期を取りながら、合成アニメーション(3)を生成する方法を示す。この場合には、最初に、位相同期点を、二つのオブジェクトの寄与率から求めた荷重和によって求める。図19Aにおいて

は、既定値を  $w : (1-w)$  として、点(4)、(5)から点(6)を求め、点(7)、(8)から点(9)を求め、点(10)、(11)から点(12)を求めている。

【0040】次に、図19Bに示すように、アニメーションの経過時間（位相）から前後の位相同期点(6)、(9)を求め、点(15)を点(6)と点(9)との内分点として表現し、これに対応する点を求める。図19Bでは、点(4)、(7)から点(13)を求め、点(5)、(8)から点(14)を求め、ここで求めた点(13)、(14)に対して、寄与率をかけることにより、点(15)に相当するフレームを求める。そして、これを続けることによって、同期アニメーションを生成する。

【0041】図20は、速度や歩幅の異なる2つの歩行から、中間の歩行パターンを持つアニメーションを生成する例と、その遷移図とを示す。

【0042】こうして、上述のアニメーションの自動生成方法によれば、立っている、歩く、跳ぶなどの個々のアニメーション（静止画を含む）を行動オブジェクトとして持ち、全体のアニメーションを、それらの行動オブジェクトを1つの状態とした状態遷移として表現するとともに、その状態遷移を行うためのきっかけはイベント（時間、割り込み、他オブジェクトからの出力、環境情報など）としている。したがって、イベントの性質などに応じて、ある状態からは自分自身も含めて複数の状態へ移行することができる。また、各状態はいつでもイベントを受け付けて他の状態に移行することができる。

【0043】さらに、各状態から別の状態に移行するときには、2つの状態におけるアニメーションを合成し、移行をスムーズになんぐアニメーションを自動的に生成するので、従来に比べて少ないデータ（キーフレームに対応させたモデルデータなど）で、より複雑なストーリーを持ち、スムーズにシーンが移行するアニメーションを作成できる。

【0044】なお、上述においては、線形補間による合成例を示したが、この他に、スプライン補間などを指定することもできる。あるいは、同期方法を編集することにより、倍周期同期や、カオスを発生させるなど、複雑な動きを生成することもできる。また、この方法を用いることにより、洋服のしわ（ドレープ）のアニメーションや髪のアニメーションを、リアルタイムの物理シミュレーションを行わずに、リアルタイムに、しかも違和感の少ない内容に生成することができる。

#### 【0045】

【発明の効果】この発明によれば、歩く・跳ぶなどの個々のアニメーションを行動オブジェクトとして保持し、全体のアニメーションはそれらの状態遷移として考え、ある行動オブジェクトから別の行動オブジェクトにはイ

ベントによって移行するとともに、各行動オブジェクトはその状態が完結していないなくても、イベントを受け付けることを可能としているので、個々のアニメーションの間をスムーズにつないだり、イベントに対するアニメーションの変化を考慮したアニメーションを自動的に生成したりすることができる。

#### 【図面の簡単な説明】

【図1】この発明を説明するための状態遷移図である。

【図2】この発明を説明するためのハードウェアのモジュール構成図である。

【図3】この発明を説明するためのモジュール構成図である。

【図4】行動オブジェクト作成の流れを示す図である。

【図5】アニメーションの設定を示す図である。

【図6】行動オブジェクト間における優先付けを示す図である。

【図7】この発明を説明するための状態遷移図である。

【図8】全体アニメーションの生成の流れを示す図である。

【図9】状態遷移の管理を示す図である。

【図10】オブジェクト間の相互作用と全体アニメーションの生成を示す図である。

【図11】オブジェクトの生成に対する生成アニメーションの決定方法を示す図である。

【図12】行動の種類と遷移アニメーションの種類を示す図である。

【図13】ポーズ間の遷移を示す図である。

【図14】ポーズからアニメーションへの遷移を示す図である。

【図15】アニメーションからアニメーションへの遷移を示す図である。

【図16】周期的アニメーション間の遷移を示す図である。

【図17】遷移アニメーションを示す図である。

【図18】遷移アニメーションを示す図である。

【図19】平行オブジェクトにおけるアニメーションの同期を示す図である。

【図20】平行オブジェクトによる歩行の合成を示す図である。

【図21】従来のアニメーションの作成方法を示す図である。

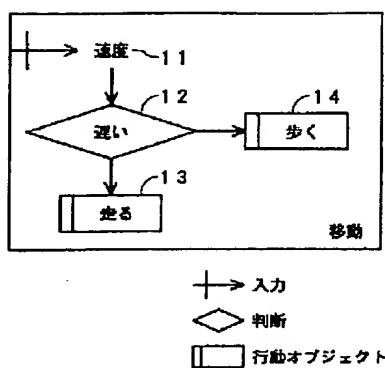
【図22】従来のアニメーションの切り換え方法を示す図である。

【図23】格闘ゲームのシーンを示す図である。

#### 【符号の説明】

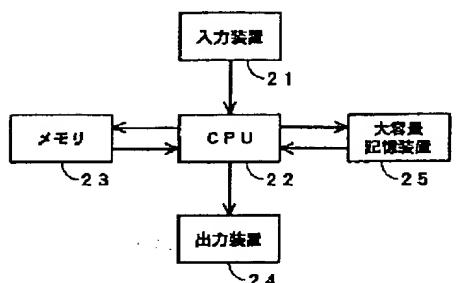
2 1…入力装置、2 2…CPU、2 3…メモリ、2 4…出力装置、2 5…大容量記憶装置

(图 1)

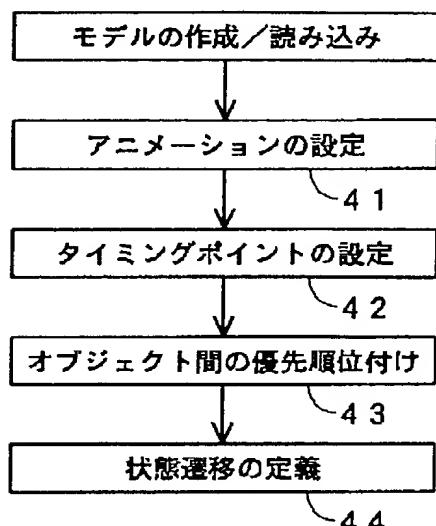


【図3】

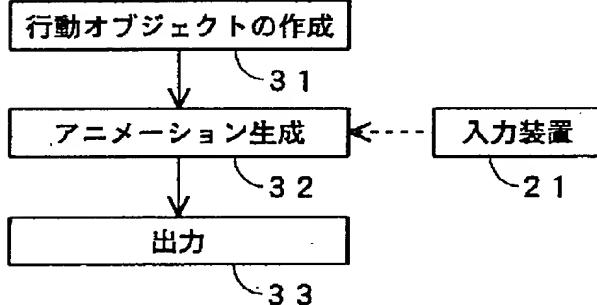
【图2】



[图4]



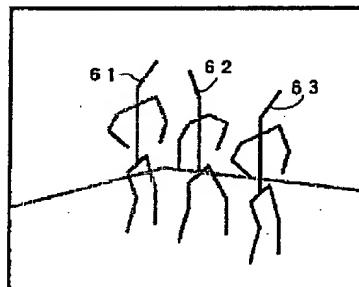
• 8 •



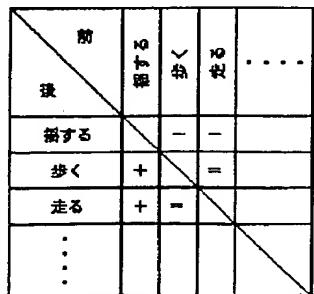
【図5】

[図6]

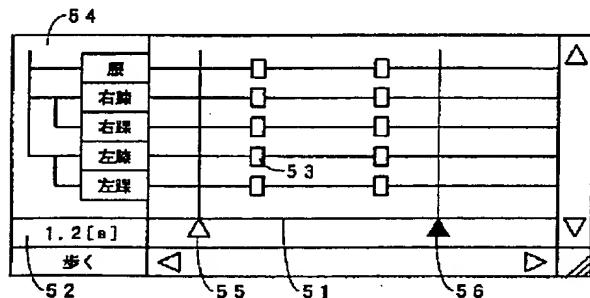
優先順位	-1	0	1	2	△
動作名		握る ・ ・ ・ ・	歩く 走る ・ ・ ・		▽
	◀			▶	△



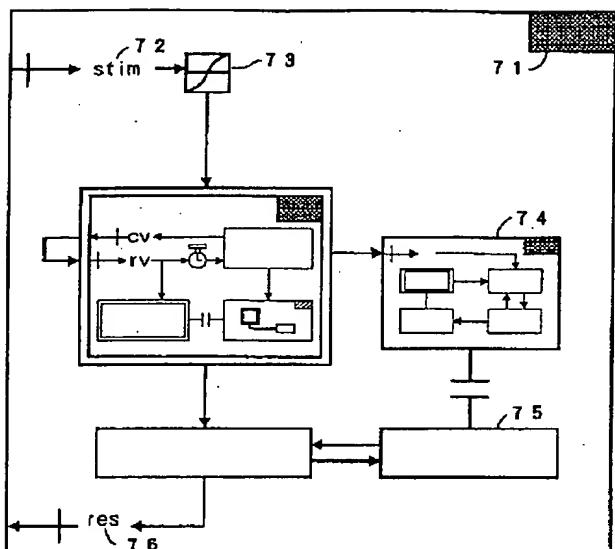
B



B

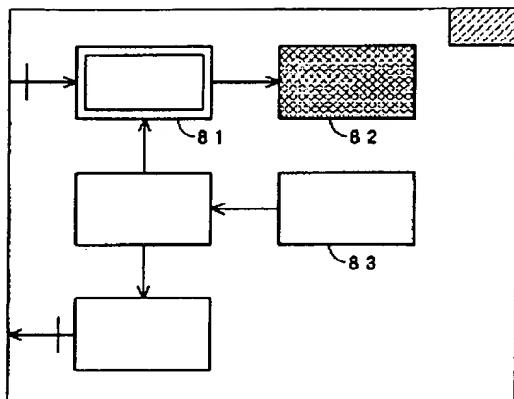


【図7】



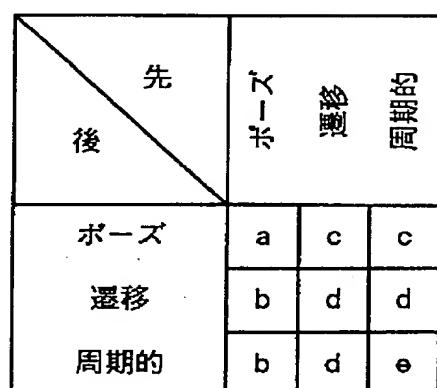
↓  
+ → 入力  
↔ → 出力  
→ → 状態遷移  
+ + → 平行関係

【図9】

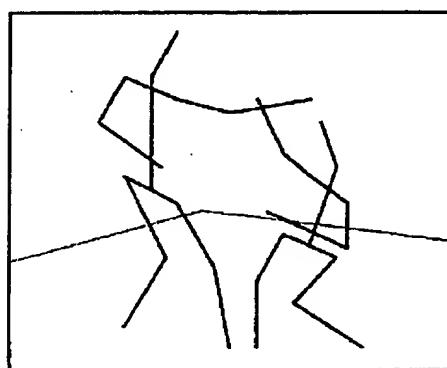


□ 遷移状態  
□ 開始状態  
▨ 現在の状態

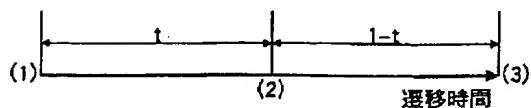
【図12】



【図23】

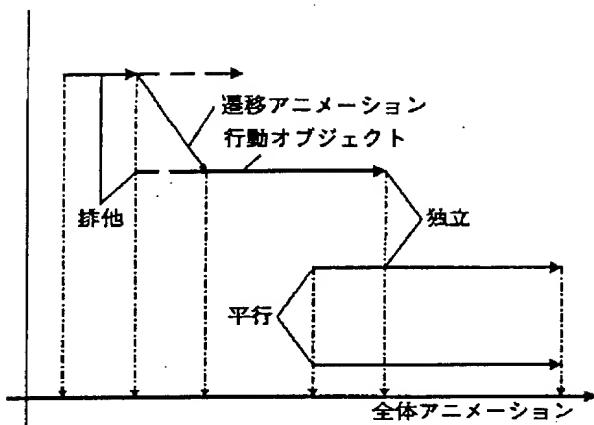


【図13】

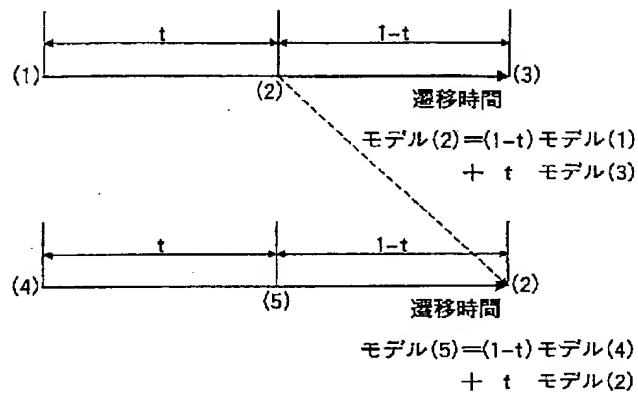


モデル(2)=(1-t) モデル(1)  
+ t モデル(3)

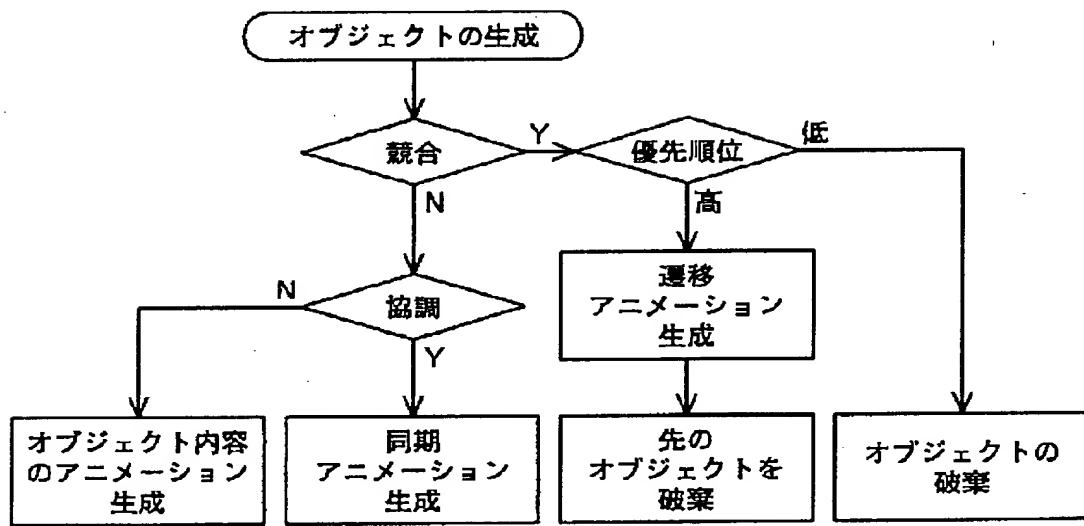
【図10】



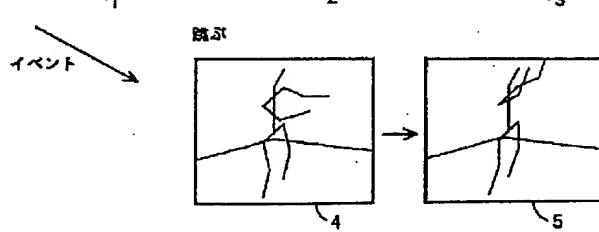
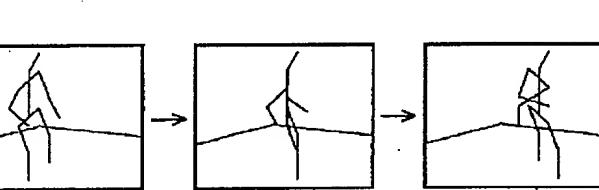
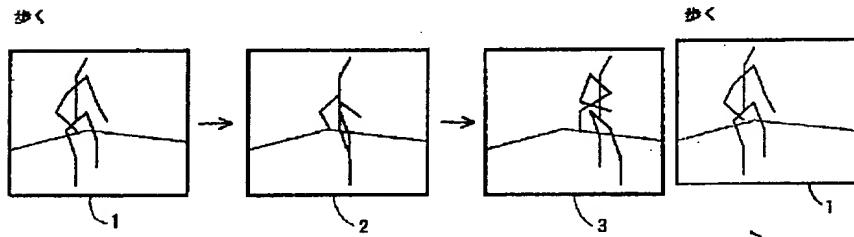
【図14】



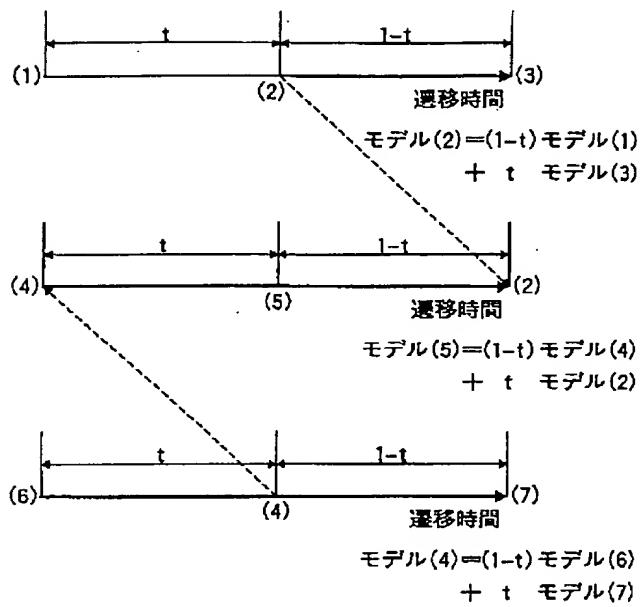
【図11】



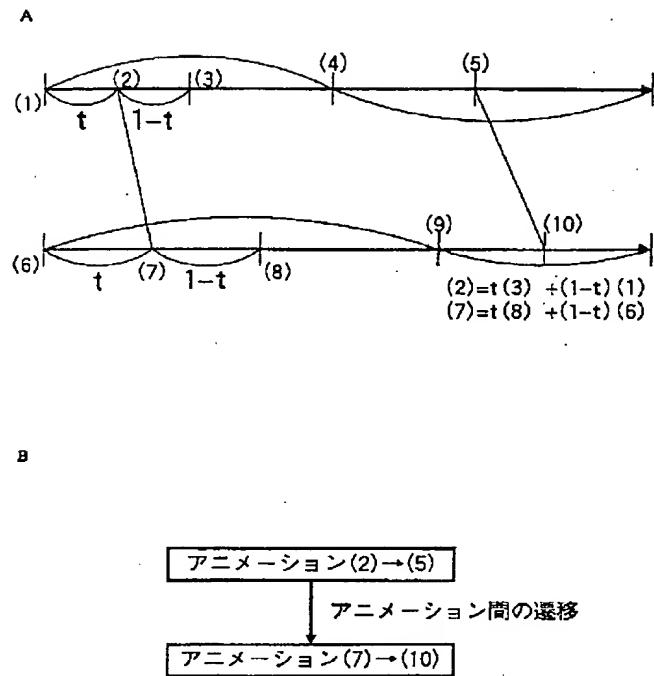
【図21】



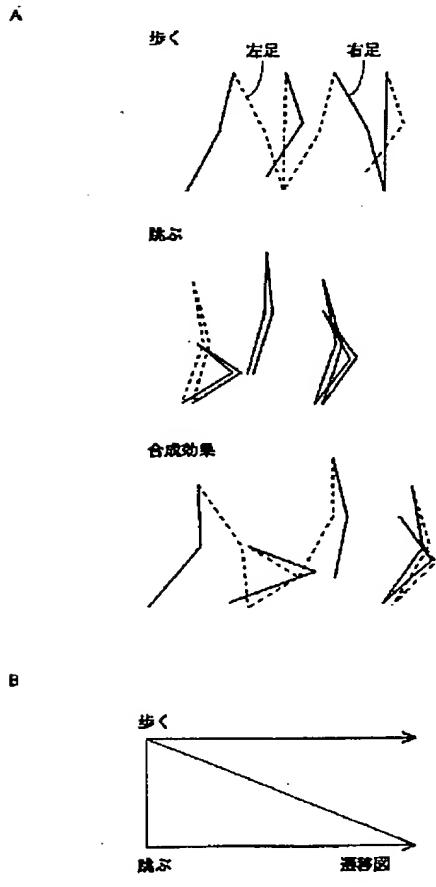
【図15】



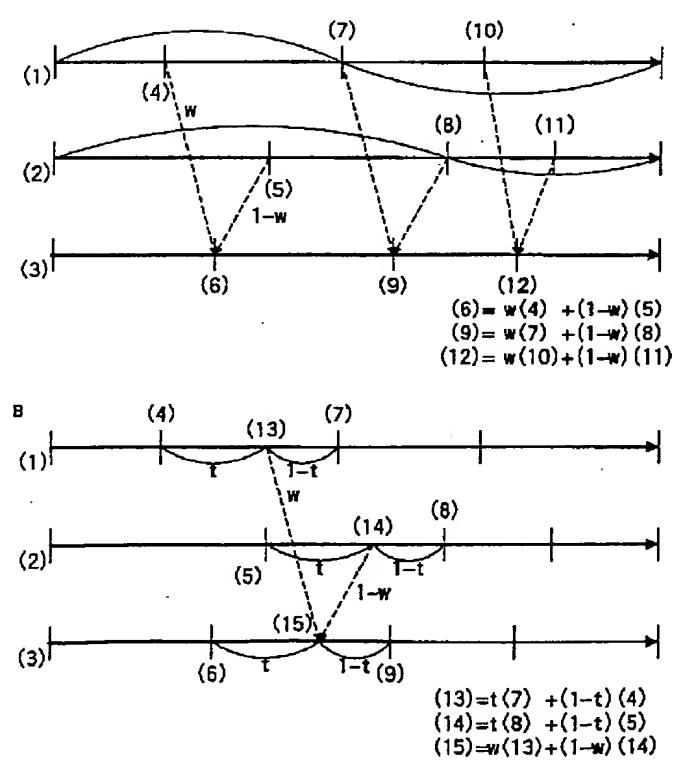
【図16】



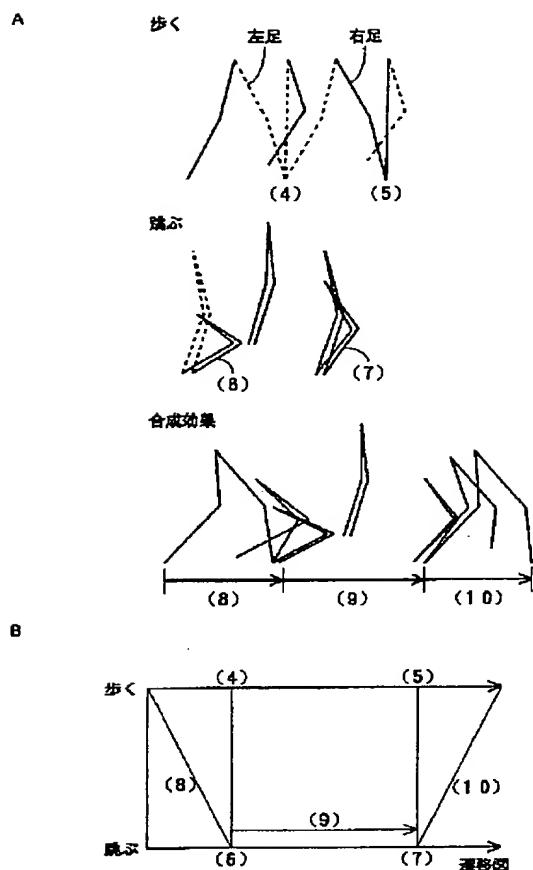
【図17】



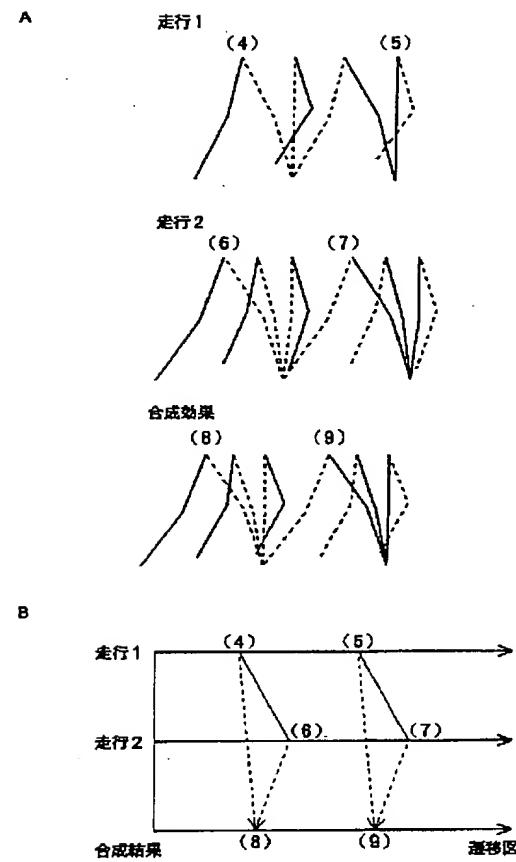
【図19】



【図18】



【図20】




---

フロントページの続き

(72)発明者 上田 裕一  
東京都品川区東五反田1丁目14番10号 株式会社ソニー木原研究所内

(72)発明者 小松 宏輔  
東京都品川区北品川6丁目7番35号 ソニーネットワークシステム株式会社内  
Fターム(参考) 5B050 BA08 BA09 DA10 EA19 EA24  
EA28